

# Charged particles constrained to a curved surface

**Thomas Müller**

Visualisierungsinstitut der Universität Stuttgart (VISUS)  
Allmandring 19, 70569 Stuttgart, Germany

E-mail: [Thomas.Mueller@visus.uni-stuttgart.de](mailto:Thomas.Mueller@visus.uni-stuttgart.de)

**Jörg Frauendiener**

Department of Mathematics & Statistics, University of Otago,  
P.O. Box 56, Dunedin 9010, New Zealand

E-mail: [joergf@maths.otago.ac.nz](mailto:joergf@maths.otago.ac.nz)

**Abstract.** We study the motion of charged particles constrained to arbitrary two-dimensional curved surfaces but interacting in three-dimensional space via the Coulomb potential. To speed-up the interaction calculations, we use the parallel compute capability of the Compute Unified Device Architecture (CUDA) of today's graphics boards. The particles and the curved surfaces are shown using the Open Graphics Library (OpenGL). The paper is intended to give graduate students, who have basic experiences with electrostatics and differential geometry, a deeper understanding in charged particle interactions and a short introduction how to handle a many particle system using parallel computing on a single home computer.

PACS numbers: 01.50.hv, 02.40.Hw, 02.60.Cb, 45.50.-j

Submitted to: *EJP*

## 1. Introduction

The original idea of the Thomson problem [1, 2] was to find equilibrium positions of  $N$  charges constrained to a spherical surface interacting with Coulomb's law. From Earnshaw's theorem [3, 4] we know, however, that a collection of point charges cannot be maintained in a stable stationary equilibrium configuration solely by their mutual electrostatic interactions. Nonetheless, there is a steady interest to find minimum energy configurations of  $N$  charged particles on a sphere, see e.g. [5, 6, 7, 8, 9, 10, 11]. There is also an interactive Java applet by Bowick et al. [12] from Syracuse university (NY) to find minimum configurations for the more general  $r^{-n}$  potential using several different minimization algorithms.

In this paper, we generalize the Thomson problem to arbitrary curved non-self-penetrating parametrized two-dimensional surfaces that are embedded in three-dimensional Cartesian space. We present the mathematical details to study the motion and minimum energy configurations of an arbitrary number of charged particles constrained to these surfaces, and we briefly describe how to implement the resulting N-body simulation using the Compute Unified Device Architecture (CUDA) of today's graphics boards. The generalization to  $r^{-n}$  potentials is left as exercise for motivated students having basic experiences with electrostatics and differential geometry.

The structure of the paper is as follows. In Sec. 2 we briefly discuss the differential geometric details for the parametrization of trajectories on curved surfaces. The equations of motion for a free particle and of particles interacting by means of the Coulomb law are described in Sec. 3. In Sec. 4 we discuss some implementation details for the N-body simulation and the subsequent visualization. Several examples and some feasible exercises are presented in sections 5 and 6.

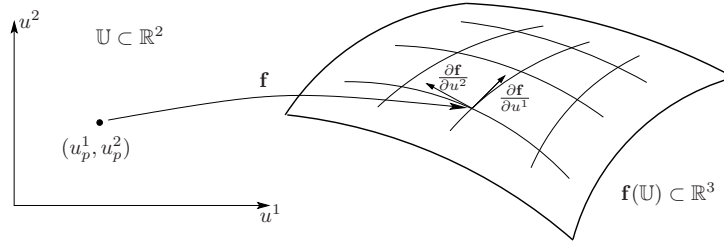
The source code to reproduce the examples in this paper is written in C/C++/CUDA and is freely available from (<http://www.vis.uni-stuttgart.de/~muelleta/ChaPaCS>). It can be compiled on Linux and Windows systems.

## 2. Particle trajectories on curved surfaces

The trajectory  $\gamma(t)$  of a particle that is constrained to a curved surface is described in a straightforward manner by means of differential geometry. For that, consider a two-dimensional surface in the common three-dimensional Cartesian space  $\mathbb{R}^3$  parametrized by the function

$$\mathbf{f}: \mathbb{R}^2 \supset \mathbb{U} \rightarrow \mathbb{R}^3, \quad (u^1, u^2) \mapsto \mathbf{f}(u^1, u^2), \quad (1)$$

see figure 1.



**Figure 1.** Parametrized surface  $\mathbf{f}(\mathbb{U}) \subset \mathbb{R}^3$ .

The intrinsic geometry of this surface is given by the first fundamental form

$$I \left( \frac{\partial \mathbf{f}}{\partial u^i}, \frac{\partial \mathbf{f}}{\partial u^j} \right) = \left\langle \frac{\partial \mathbf{f}}{\partial u^i}, \frac{\partial \mathbf{f}}{\partial u^j} \right\rangle = g_{ij} \quad (2)$$

which defines a metric  $(\mathbf{g})_{ij} = g_{ij}$  on  $\mathbf{f}$ . Here,  $\langle \cdot, \cdot \rangle$  denotes the standard scalar product in  $\mathbb{R}^3$ . The corresponding Christoffel symbols of the first kind

$$2\Gamma_{ij,k} = \frac{\partial g_{jk}}{\partial u^i} + \frac{\partial g_{ik}}{\partial u^j} - \frac{\partial g_{ij}}{\partial u^k} \quad (3)$$

describe the effects of parallel transport in curved spacetime. Later on, we also need the Christoffel symbols of the second kind which follow from the Christoffel symbols of the first kind by raising the last index using the inverse metric  $g^{ij} = (\mathbf{g}^{-1})_{ij}$ ,

$$\Gamma_{ij}^k = \sum_{m=1}^2 g^{km} \Gamma_{ij,m} \quad (4)$$

The inverse of the  $2 \times 2$  matrix  $(\mathbf{g})_{ij} = g_{ij}$  reads

$$\mathbf{g}^{-1} = \frac{1}{\det(\mathbf{g})} \begin{pmatrix} g_{22} & -g_{12} \\ -g_{21} & g_{11} \end{pmatrix} \quad (5)$$

with determinant  $\det(\mathbf{g}) = g_{11}g_{22} - g_{12}g_{21}$ .

A particle trajectory on  $\mathbf{f}$  can now be represented by the parametrization  $\gamma: \vec{u}(t) = (u^1(t), u^2(t)) \in \mathbb{U}$  for  $t \in [t_i, t_f]$ , where  $t$  usually denotes time. The time derivative of  $\mathbf{f}$  yields the tangent of  $\gamma$  in  $\mathbb{R}^3$ ,

$$\dot{\mathbf{f}} = \frac{d}{dt} \mathbf{f}(\vec{u}(t)) = \sum_{i=1}^2 \frac{\partial \mathbf{f}}{\partial u^i} \frac{du^i}{dt} = \sum_{i=1}^2 \frac{\partial \mathbf{f}}{\partial u^i} \dot{u}^i. \quad (6)$$

The scalar product  $\dot{\mathbf{f}}^2 = \langle \dot{\mathbf{f}}, \dot{\mathbf{f}} \rangle$  can be represented using the first fundamental form:

$$\dot{\mathbf{f}}^2 = \left\langle \frac{\partial \mathbf{f}}{\partial u^1}, \frac{\partial \mathbf{f}}{\partial u^1} \right\rangle (\dot{u}^1)^2 + 2 \left\langle \frac{\partial \mathbf{f}}{\partial u^1}, \frac{\partial \mathbf{f}}{\partial u^2} \right\rangle \dot{u}^1 \dot{u}^2 + \left\langle \frac{\partial \mathbf{f}}{\partial u^2}, \frac{\partial \mathbf{f}}{\partial u^2} \right\rangle (\dot{u}^2)^2 = \sum_{i,j=1}^2 g_{ij} \dot{u}^i \dot{u}^j \quad (7)$$

Several examples of surface parametrizations are listed in Appendix D.

### 3. Particle motion

#### 3.1. Equation of motion for free particles

As long as there is only a single particle or there is no mutual interaction or external force, the trajectory of a single particle of mass  $m$  on the surface  $\mathbf{f}$  follows from the Euler-Lagrangian (EL) equations

$$0 = \frac{d}{dt} \frac{\partial L}{\partial \dot{u}^k} - \frac{\partial L}{\partial u^k}, \quad k = \{1, 2\} \quad (8)$$

with Lagrangian  $L = \frac{m}{2} \dot{\mathbf{f}}^2$  and  $\dot{\mathbf{f}}^2$  from (7). The derivative of the Lagrangian  $L$  with respect to  $\dot{u}^k$  and the subsequent time derivative yields the first part of the EL equations,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{u}^k} = m \left[ \sum_{i=1}^2 \left( \frac{d}{dt} g_{ik} \right) \dot{u}^i + \sum_{i=1}^2 g_{ik} \ddot{u}^i \right]. \quad (9)$$

For the partial derivative of  $L$  with respect to  $u^k$  we obtain

$$\frac{\partial L}{\partial u^k} = \frac{m}{2} \sum_{i,j=1}^2 \frac{\partial g_{ij}}{\partial u^k} \dot{u}^i \dot{u}^j. \quad (10)$$

The derivatives of the metric  $g_{ij}$  are detailed in Appendix B. Equations (9) and (10) contracted with the inverse metric  $g^{ij}$  lead to the geodesic equation

$$0 = \ddot{u}^k + \sum_{i,j=1}^2 \Gamma_{ij}^k \dot{u}^i \dot{u}^j \quad (11)$$

of a free particle constrained onto  $\mathbf{f}$ . As expected, the mass  $m$  of the particle drops out. In flat space,  $g_{ij} = \delta_{ij}$  with  $\delta_{ij}$  being the Kronecker- $\delta$ , the Christoffel symbols vanish and (11) represents the parametrized form of the force-free Newtonian equation.

#### 3.2. Equation of motion for charged particles

The Lagrangian for a charged particle  $q$  under the influence of the Coulomb interaction of  $N$  other charged particles located at  $\mathbf{f}_n$  reads

$$L = \frac{m}{2} \dot{\mathbf{f}}^2 - \sum_{n=1}^N \frac{qq_n}{4\pi\epsilon_0} \frac{1}{\|\mathbf{f} - \mathbf{f}_n\|} \quad (12)$$

with the distance  $r := \|\mathbf{f} - \mathbf{f}_n\| = \sqrt{\langle \mathbf{f}(\vec{u}) - \mathbf{f}(\vec{u}_n), \mathbf{f}(\vec{u}) - \mathbf{f}(\vec{u}_n) \rangle}$  between a particle at  $\mathbf{f}(\vec{u})$  and a particle at  $\mathbf{f}(\vec{u}_n)$ . For the Euler-Lagrangian equations we need the partial derivative of the reciprocal distance

$$\frac{\partial}{\partial u^j} r^{-1} = -\frac{1}{2} r^{-3} \frac{\partial}{\partial u^j} \langle \mathbf{f} - \mathbf{f}_n, \mathbf{f} - \mathbf{f}_n \rangle = -r^{-3} \left\langle \mathbf{f} - \mathbf{f}_n, \frac{\partial \mathbf{f}}{\partial u^j} \right\rangle. \quad (13)$$

As  $r^{-1}$  is independent of  $\dot{u}^j$ , the equation of motion of a charged particle under the influence of  $N$  other charged particles on  $\mathbf{f}$  is given by

$$0 = \ddot{u}^k + \sum_{i,j=1}^2 \Gamma_{ij}^k \dot{u}^i \dot{u}^j + \sum_{n=1}^N \frac{qq_n}{4\pi\epsilon_0} \sum_{j=1}^2 \frac{1}{r^3} \left\langle \mathbf{f} - \mathbf{f}_n, \frac{\partial \mathbf{f}}{\partial u^j} \right\rangle g^{jk} \quad (14)$$

### 3.3. Kinetic and field energy

The total kinetic energy  $T$  of all charged particles is just the sum of the kinetic energies of the single particles:

$$T = \frac{m}{2} \sum_{i=1}^N \dot{\mathbf{f}}_i^2. \quad (15)$$

The field energy  $W$  follows from

$$W = \frac{1}{4\pi\epsilon_0} \sum_{i=2}^N \sum_{j=1}^{i-1} \frac{q_i q_j}{\|\mathbf{f}_i - \mathbf{f}_j\|}. \quad (16)$$

### 3.4. External electric and magnetic fields

The influence of an external electric or magnetic field onto a single charged particle on a curved surface is expressed by the Lagrangian

$$L = \frac{m}{2} \dot{\mathbf{f}}^2 - q\phi(\mathbf{f}, t) + q \langle \dot{\mathbf{f}}, \mathbf{A}(\mathbf{f}, t) \rangle \quad (17)$$

with the electric potential  $\phi$ , the magnetic vector potential  $\mathbf{A}$ , and the relation  $\mathbf{E} = -\nabla\phi$  and  $\mathbf{B} = \nabla \times \mathbf{A}$ . The Euler-Lagrangian equations yield

$$0 = \ddot{u}^k + \sum_{i,j=1}^2 \Gamma_{ij}^k \dot{u}^i \dot{u}^j - q \left\langle \mathbf{E} + \dot{\mathbf{f}} \times \mathbf{B}, \frac{\partial \mathbf{f}}{\partial u^j} \right\rangle g^{jk} + q \left\langle \frac{\partial \mathbf{f}}{\partial u^j}, \frac{\partial \mathbf{A}}{\partial t} \right\rangle g^{jk}. \quad (18)$$

If the magnetic vector potential does not explicitly depend on time  $t$ , the last term of (18) can be dropped.

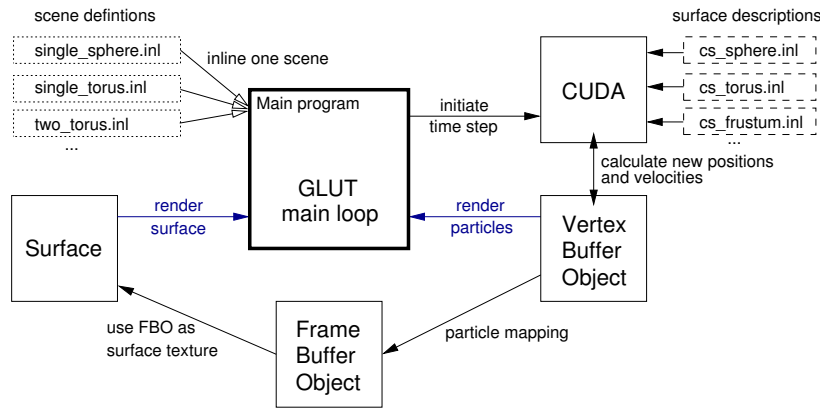
## 4. N-body simulation and visualization

As long as the number of particles in an N-body simulation is in the order of a few thousands, we do not need any specific acceleration algorithm and/or approximation procedure, but we can calculate the N-body interaction by brute force: every particle interacts with every other particle. The computation time, however, increases quadratically with the number of particles. A first step to accelerate the computation is to handle each particle by a separate compute unit that has to integrate the equation of motion (14) for this particle. The only prerequisite is that all compute units must have access to all particle positions and velocities which can be achieved using a shared memory system. Today, virtually all standard home computers and even high-end smartphones have at least a dual core processor inside that have access

to shared memory. Parallelization of computation can then be realized, for example, by the programming interface *OpenMP*[13] that splits the computation into several threads.

Much higher parallelization can be achieved using the compute capability of modern graphics hardware. The *Compute Unified Device Architecture* (CUDA) or the *Open Compute Language* (OpenCL) offer a C-like programming interface in order to use the graphics processing units (GPUs) for general purpose computations. Even without sophisticated algorithms for an efficient memory access, GPU computation leads to an enormous speed-up. Together with the Open Graphics Library (OpenGL) we can explore physical simulations at interactive frame rates.

The basic structure of our implementation is shown in figure 2. The basic block is the GLUT [14] main loop which acts on key strokes and mouse events, and which initiates rendering new frames. The “CUDA” block includes all surface descriptions ( $\mathbf{f}, \partial\mathbf{f}, g_{ij}, \Gamma_{ij}^k$ )



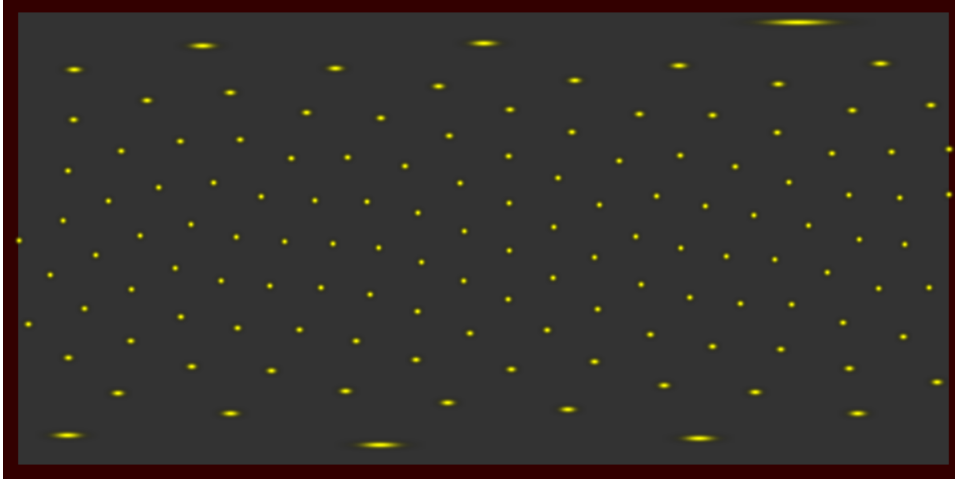
**Figure 2.** Basic structure of the program.

and calculates a single time step using either a standard Runge-Kutta second or fourth order method, see e.g. Press et al. [15]. To calculate a time step, it uses the particle positions  $u^k$  and velocities  $u^k$  stored within a *Vertex Buffer Object* (VBO) which can be directly accessed by CUDA and OpenGL. The new particle positions can then be rendered directly or they can be mapped onto their corresponding surfaces. This mapping is realized by means of a *Frame Buffer Object* (FBO). This FBO is an internal rectangular image (display) which, in our case, represents the domain  $\mathbb{U}$ . This image is then used to texturize the surface.

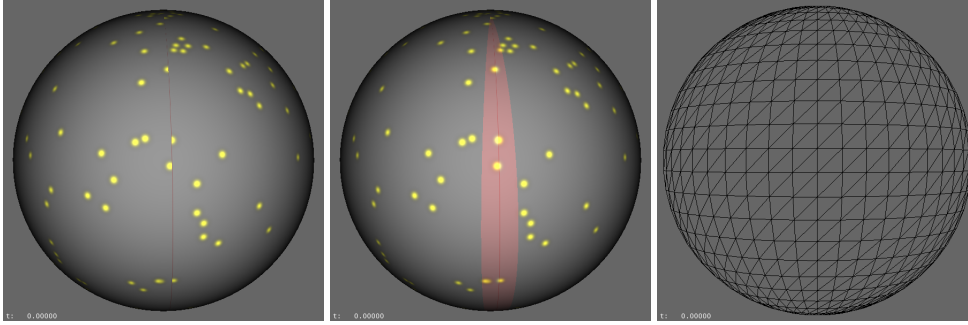
As an example, figure 3 shows particles (yellow splats) projected onto a rectangular image (FBO) that is used as texture for the sphere. The inner gray rectangle covers the whole domain of the sphere  $u^1 = \varphi \in [0, 2\pi)$ ,  $u^2 = \vartheta \in (0, \pi)$ . The dark red border slightly expands the domain to prevent particle splats near the domain’s boundary from being clipped, see figure 4. The splats have to be distorted by means of the inverse metric of the surface to let them appear as circular splats when mapped onto the surface.

The “Surface” block is responsible for drawing the surfaces themselves. For that, we uniformly sample the domain using quads. These quads are then split into two triangles which are transformed by means of the surface function  $\mathbf{f}$  of (1) within a so called vertex shader (see OpenGL Shading Language [16]). Figure 4 (right) shows the resulting wireframe of a sphere.

To keep the code simple, we do not have any sophisticated scene description language but implement each scene in a separate “.inl”-file. A specific scene and its particular scene parameters have to be chosen at compile time. Each “.inl”-file must have three functions:



**Figure 3.** Particle rendering as Gaussian splats with radii  $d\vartheta = ds/r$  and  $d\phi = ds/(r\sin\vartheta)$  for fixed value  $ds$ . This FBO image can be mapped onto a sphere, see figure 4. The inner gray rectangle covers the whole domain of the sphere  $u^1 = \phi \in [0, 2\pi)$ ,  $u^2 = \vartheta \in (0, \pi)$ . The dark red border (color online) prevents particle splats near the domain's boundary from being clipped.



**Figure 4.** Particle rendering as Gaussian splats without (left) and with (center) extended domain, compare figure 3. Without extended domain the splats are clipped. Right: wireframe view of the sphere.

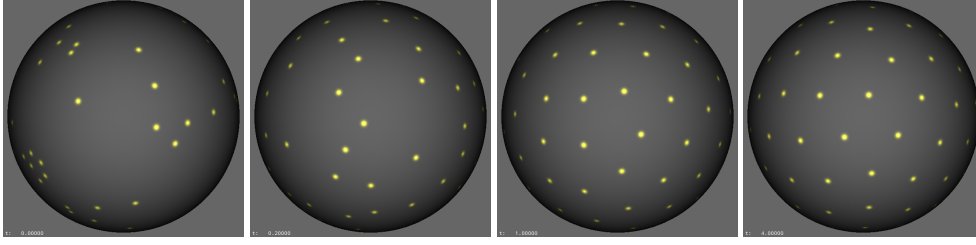
`init_Objects()` defines all surfaces and assigns IDs to them; `set_Particles()` registers the number of particles, their parameters (mass, charge, initial position, initial velocity), and the ID of the surface they belong to; `set_Supplement()` offers the possibility to change the camera parameters or the size of the window. The particle data could also be loaded from file when starting the program.

## 5. Examples

In the following examples, we use the explicit second-order Runge-Kutta method without step size control to integrate the equation of motion (14). The units in use are explained in Appendix A.

As a first example, we consider a Thomson problem situation where  $N = 128$  charged particles ( $q = 10e$ ) are located on a sphere of radius  $r = 1 \text{ mm}$ . At the beginning of the

simulation, the particles are randomly distributed and have zero initial velocity, see figure 5. The field energy  $W|_{t=0}/(\kappa m_e) \approx 8.66684 \cdot 10^5 \text{ mm}^{-1}$  and the kinetic energy  $T = 0$ . To obtain a minimum energy configuration, we add a linear friction term with frictional constant  $\eta = 10$ , see Appendix C. At simulation time ( $t = 4s$ ), the field energy has dropped to  $W|_{t=4}/(\kappa m_e) \approx 7.39316 \cdot 10^5 \text{ mm}^{-1}$  and the kinetic energy has reduced to  $T/m_e \approx 8.11308 \cdot 10^{-3} \text{ mm}^2 \text{ s}^{-2}$ , which is nearly impossible to observe visually, however. After about 19.5 seconds, the kinetic energy has dropped below  $T/m_e = 10^{-12} \text{ mm}^2 \text{ s}^{-2}$ .



**Figure 5.**  $N = 128$  charged particles on a sphere at simulation times  $t = \{0, 0.2, 1, 4\} s$  with step size  $\Delta t = 10^{-3} s$  and velocity-dependent friction  $\eta = 10$  (see Appendix C).

In table 1 we compare our minimum energy results for several number of particles  $N$  with the literature values. Glasser and Every's [17] estimation formula

$$E(N) = \frac{N^2}{2} \left( 1 - aN^{-1/2} + bN^{-3/2} \right) \quad (19)$$

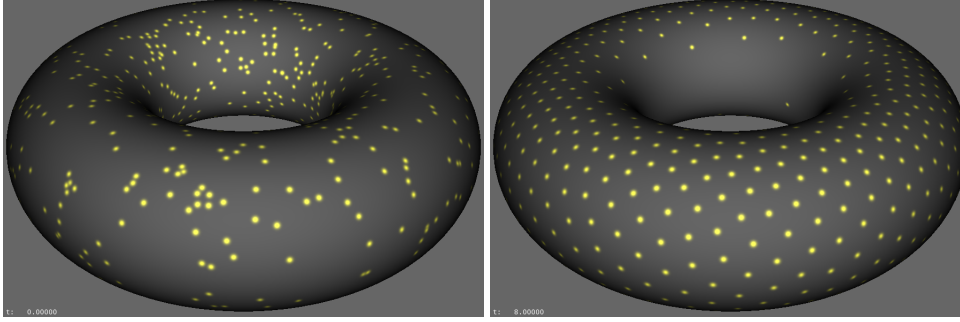
with parameters  $a = 1.10461$  and  $b = 0.137$  from [18] already gives a good approximation. The lowest energies for  $110 < N \leq 200$  from Morris et al. [18] are determined using a genetic algorithm.

$N$	$W$	$100W_N$	$100E(N)$
112	$5.618044887 \cdot 10^5$	$5.618044882 \cdot 10^5$	$5.618079704 \cdot 10^5$
128	$7.393007455 \cdot 10^5$	$7.393007443 \cdot 10^5$	$7.392951914 \cdot 10^5$
161	$1.183308476 \cdot 10^6$	$1.183308474 \cdot 10^6$	$1.183308683 \cdot 10^6$
200	$1.843885657 \cdot 10^6$	$1.843884272 \cdot 10^6$	$1.843881429 \cdot 10^6$

**Table 1.** Minimum field energies  $W$  for  $N$  particles with charge  $q = 10e$ , time step  $\Delta t = 10^{-3} s$  and frictional constant  $\eta = 10$ ; compared with values  $W_N$  taken from Morris et al. [18] and estimation given by (19). The factor 100 is because we use  $q = 10e$  instead of  $q = 1e$ .

In the second example, we consider  $N = 1024$  particles ( $q = 10e$ ) on a torus with radii  $R = 2 \text{ mm}$  and  $r = 0.9 \text{ mm}$ . As before, the particles are randomly distributed at the beginning of the simulation, see figure 6. At  $t \approx 30s$ , the kinetic energy has dropped below  $T/m_e = 10^{-8} \text{ mm}^2 \text{ s}^{-2}$  and the field energy reads  $W/(\kappa m_e) \approx 2.19652 \cdot 10^7 \text{ mm}^{-1}$ . Expectedly, all particles move to the outer side of the torus. Again, to stress the validity of our code, we compare our minimum energy configurations with the literature, see table 2. As can be seen, we are in good agreement with the literature values and sometimes we have found even lower energies.

The dynamic evolution of the  $N = 415$  example of table 2 can be read from figure 7. At the beginning of the simulation, there is a very short peak of high kinetic energy due to the particles that are accelerated from rest. After less than 2 seconds, the kinetic energy has

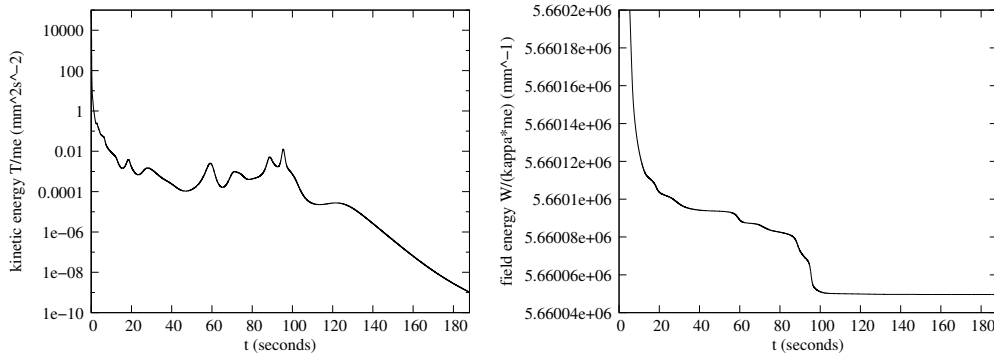


**Figure 6.**  $N = 1024$  charged particles on a torus at simulation times  $t = \{0, 8\}s$  with step size  $\Delta t = 10^{-2}s$  and velocity-dependent friction  $\eta = 200$ .

$N$	$a = R/r$	$W$	$100W_N$
20	1.414	$1.029846718 \cdot 10^4$	$1.029846689 \cdot 10^4$
20	1.618	$1.098582566 \cdot 10^4$	$1.098582529 \cdot 10^4$
100	1.414	$3.082218338 \cdot 10^5$	$3.082217005 \cdot 10^5$
100	1.618	$3.295927689 \cdot 10^5$	$3.296043624 \cdot 10^5$
415	1.414	$5.660049622 \cdot 10^6$	$5.660070457 \cdot 10^6$

**Table 2.** Minimum field energies  $W$  for  $N$  particles with charge  $q = 10e$ , time step  $\Delta t = 0.01 s$  and frictional constant  $\eta = 50$  on a torus with aspect  $a = R/r$  and radius  $R = 1$ ; compared with values  $W_N$  taken from [12].

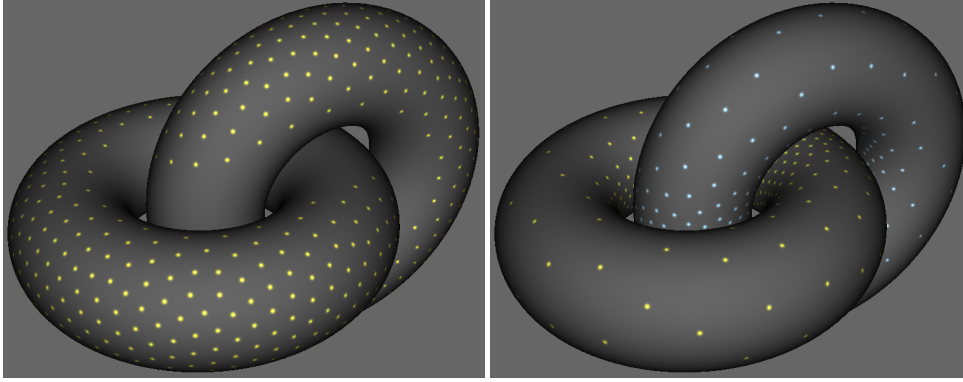
dropped below  $T/m_e = 1 \text{ mm}^2 s^{-2}$ . Then, at least from the visual impression, the particles do not move any more. However, there is still kinetic energy in the system which does not dissipate uniformly. Only after about 100 seconds, the particles slowly settle down and reach a minimum energy configuration (plateau in figure 7, right). However, depending on the initial random configuration, the simulation does not always reach the same minimum energy configuration. Then, the particles have to be either randomly distributed again or they have to be given a small jerk by pressing a key.



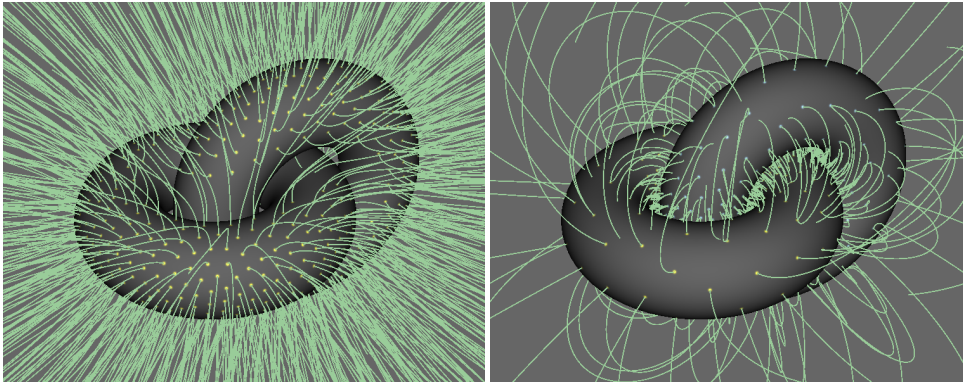
**Figure 7.** Kinetic energy (left) and field energy (right) depending on time for the torus simulation with  $N = 415$  particles, aspect  $a = 1.414$ , step size  $\Delta t = 0.01$ , and a frictional constant  $\eta = 50$ . The loss of kinetic energy is also due to the dissipative RK2 method.



A more intricate example is shown in figure 8 where two torii are intertwined and the particles have either the same or opposite charges. The corresponding field lines are shown in figure 9. They were started close to the charged particles in the direction of the outer surface normal and were integrated with a constant step size along the total electric field of all particles. Please note that the field lines end after  $n = 800$  steps by default. And, as the object surface is no equipotential surface, field lines could also penetrate the object surface non-orthogonally.



**Figure 8.**  $N = 2 \times 512$  particles ( $q = \pm 10e$ ) on two intertwined torii with radii  $R = 2 \text{ mm}$  and  $r = 0.9 \text{ mm}$  of either the same (left) or opposite (right) charge.



**Figure 9.** The same situation as in figure 8 but now with green lines which represent field lines that were started close to the particle positions and perpendicular to the surface.

## 6. Exercises

In the following we give a few suggestions for possible exercises that could be done using the ChaPaCS source code.

- Consider the Thomson problem situation where  $N = 128$  charged particles ( $q = 10e$ ) are located on a sphere. How does the minimum field energy vary with the size of the radius?
- Adapt the `set_Particles()` method of the single sphere example such that one half of the particles have  $q = 10e$  and the other ones have only  $q = 5e$ . What happens?

- Determine the influence of the aspect in the torus example on the minimum field energy.
- Find the parametrization of an ellipsoid and determine the metric coefficients  $g_{ij}$  and the Christoffel symbols  $\Gamma_{ij}^k$ . Expand the code to handle ellipsoids and find the minimum energy configurations.
- Construct a new scene with a small sphere hovering above a plane.
- Construct a new scene with four small spheres at the corners of a quad (quadrupole). Study the distribution of particles on the spheres and the overall field lines.
- Determine a minimum energy configuration for  $N = 128$  charged particles ( $q = 10e$ ) on the sphere and save it to disk. Restart the program with this configuration. Set the frictional constant to zero and add an external electric field. Slightly increase the electric field strength and discuss what happens.

## 7. Summary

In this work we have developed the equations of motion of  $N$  charged particles that are constrained to a curved two-dimensional surface but interact in three dimensions as usual. We have also given a short introduction how to implement the particle simulation using the compute capability of today's graphics boards. The source code (C/C++/CUDA) of the prototype implementation *ChaPaCS* is freely available and can be easily extended by other curved surfaces.

## Appendix A. Units

For numerical computations we should know which order of magnitudes we have to deal with. If we use the electron mass  $m_e$  and the electron charge  $e$  as basis units, we could rearrange (12) to

$$\frac{L}{m_e} = \frac{M}{2} \dot{\mathbf{f}}^2 - \frac{e^2}{4\pi\epsilon_0 m_e} \sum_{n=1}^N \frac{Q Q_n}{\|\mathbf{f} - \mathbf{f}_n\|} = \frac{M}{2} \dot{\mathbf{f}}^2 - \kappa \sum_{n=1}^N \frac{Q Q_n}{\|\mathbf{f} - \mathbf{f}_n\|}, \quad (\text{A.1})$$

where  $M = m/m_e$  and  $Q = q/e$  are dimensionless. If distances are given in millimeters and time is given in seconds, the constant  $\kappa$  reads

$$\kappa = \frac{e^2}{4\pi\epsilon_0 m_e} = \frac{e^2 \mu_0 c^2}{4\pi m_e} \approx 0.253 \frac{\text{mm}^3}{\text{s}^2}, \quad (\text{A.2})$$

where  $m_e \approx 9.109 \cdot 10^{-31} \text{ kg} \approx 5.686 \cdot 10^{-18} \text{ eV} \cdot \text{s}^2/\text{mm}^2 = 5.686 \text{ aeV} \cdot \text{s}^2/\text{mm}^2$ ,  $e \approx 1.602 \cdot 10^{-19} \text{ C}$ ,  $\mu_0 = 4\pi 10^{-7} \text{ Vs/(Am)}$ , and  $c = 299792458 \text{ m/s}$ . Furthermore, we have  $\kappa m_e \approx 1.440 \cdot 10^{-18} \text{ eV} \cdot \text{mm} = 1.440 \text{ aeV} \cdot \text{mm}$ . Then, the field energy (16), for example, reads

$$\hat{W} := \frac{W}{\kappa m_e} = \sum_{i=2}^N \sum_{j=1}^{i-1} \frac{(q_i/e)(q_j/e)}{\|\mathbf{f}_i - \mathbf{f}_j\|} \quad (\text{A.3})$$

and has dimension one over length.

## Appendix B. Derivatives of the metric

The partial derivative of  $g_{ij}$  with respect to the coordinate  $u^k$  is given by

$$\frac{\partial}{\partial u^k} g_{ij} = \frac{\partial}{\partial u^k} \left\langle \frac{\partial \mathbf{f}}{\partial u^i}, \frac{\partial \mathbf{f}}{\partial u^j} \right\rangle = \left\langle \frac{\partial^2 \mathbf{f}}{\partial u^k \partial u^i}, \frac{\partial \mathbf{f}}{\partial u^j} \right\rangle + \left\langle \frac{\partial \mathbf{f}}{\partial u^i}, \frac{\partial^2 \mathbf{f}}{\partial u^k \partial u^j} \right\rangle. \quad (\text{B.1})$$

This can also be written using the Christoffel symbols of the first kind:

$$\frac{\partial}{\partial u^k} g_{ij} = \Gamma_{ik,j} + \Gamma_{jk,i}. \quad (\text{B.2})$$

The time derivative of the first fundamental form reads

$$\frac{d}{dt} g_{ik} = \frac{\partial g_{ik}}{\partial u^n} \dot{u}^n = (\Gamma_{in,k} + \Gamma_{kn,i}) \dot{u}^n. \quad (\text{B.3})$$

The partial derivative of the inverse metric  $g^{nm}$  can be derived from  $\partial_{u^k} g_{ij} = \partial_{u^k} (g_{in} g_{jm} g^{nm})$ . With the property  $g_{in} g^{im} = \delta_n^m$  and some index juggling, we obtain

$$\frac{\partial}{\partial u^k} g^{nm} = -(\Gamma_{ik}^m g^{in} + \Gamma_{ik}^n g^{im}). \quad (\text{B.4})$$

## Appendix C. Particle motion with friction

Let the particle motion be damped by a velocity-dependent friction  $\mathbf{F}_R = -h(v) \frac{\mathbf{v}}{v}$  with the frictional function  $h$  depending on the value of the velocity  $v = \|\mathbf{v}\|$  and  $\mathbf{v} = \dot{\mathbf{f}}$ . Then, the generalized friction  $R_k$  reads

$$R_k = -h(v) \frac{\mathbf{v}}{v} \cdot \frac{\partial \mathbf{r}}{\partial u^k} = -h(v) \frac{\mathbf{v}}{v} \cdot \frac{\partial \mathbf{v}}{\partial \dot{u}^k} = -\frac{h(v)}{v} \left\langle \dot{\mathbf{f}}, \frac{\partial \dot{\mathbf{f}}}{\partial \dot{u}^k} \right\rangle \quad (\text{C.1})$$

For the linear friction  $h(v) = \eta v$  with frictional constant  $\eta$  and the time derivative of  $\mathbf{f}$ , compare (6), we obtain

$$R_k = -\eta \sum_{j=1}^2 \left\langle \frac{\partial \mathbf{f}}{\partial u^j}, \frac{\partial \mathbf{f}}{\partial u^k} \right\rangle \dot{u}^j = -\eta \sum_{j=1}^2 g_{jk} \dot{u}^j. \quad (\text{C.2})$$

The Euler-Lagrange equation (8) now reads

$$0 = \frac{d}{dt} \frac{\partial L}{\partial \dot{u}^k} - \frac{\partial L}{\partial u^k} - R_k, \quad (\text{C.3})$$

and the equation of motion (11) is given by

$$0 = \ddot{u}^k + \eta \dot{u}^k + \sum_{i,j=1}^2 \Gamma_{ij}^k \dot{u}^i \dot{u}^j. \quad (\text{C.4})$$

A friction that is quadratic in the velocity yields the term  $\eta \dot{u}^k \|\dot{\mathbf{f}}\|$  with  $\|\dot{\mathbf{f}}\| = \sqrt{\sum_{i,j=1}^2 g_{ij} \dot{u}^i \dot{u}^j}$ .

The unit of  $\eta$  depends on the unit of the coordinates  $u^k$ . As these could be different like in the frustum case, where  $[u^1 = \phi] = \text{rad}$  and  $[u^2 = z] = \text{length}$  (see Appendix D.4), we should use two separate frictional constants. But here, we use the same numerical value and use the respective unit if necessary.

### Appendix D. Surface examples

The following surface parametrizations are given in standard form, which means that the components of  $\mathbf{f}$  are with respect to the global Cartesian coordinate system of  $\mathbb{R}^3$ . Thus,  $\mathbf{f} = f^1 \mathbf{e}_1 + f^2 \mathbf{e}_2 + f^3 \mathbf{e}_3$  with

$$\mathbf{e}_1 = (1, 0, 0)^T, \quad \mathbf{e}_2 = (0, 1, 0)^T, \quad \mathbf{e}_3 = (0, 0, 1)^T, \quad (\text{D.1})$$

and the center of the object equals the origin of the global coordinate system. However, for more elaborate scenes, the basis  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  can be oriented and translated arbitrarily.

#### Appendix D.1. Plane

A plane is definitely the most simple surface and can be parametrized by the coordinates ( $u^1 = x, u^2 = y$ ) which, in general, are bound to a fixed domain  $x \in [x_a, x_b]$  and  $y \in [y_a, y_b]$ . The surface reads  $\mathbf{f}(x, y) = (x, y, 0)^T$ , and the metric is given by  $g_{11} = g_{22} = 1$  and  $g_{12} = 0$ . The Christoffel symbols of the second kind vanish identically.

#### Appendix D.2. Sphere

The surface of a sphere with radius  $r$  can be parametrized by spherical coordinates ( $u^1 = \varphi, u^2 = \vartheta$ ):

$$\mathbf{f}(\varphi, \vartheta) = r \begin{pmatrix} \sin \vartheta \cos \varphi \\ \sin \vartheta \sin \varphi \\ \cos \vartheta \end{pmatrix} \quad (\text{D.2})$$

with  $\varphi \in [0, 2\pi)$ ,  $\vartheta \in (0, \pi)$ , and partial derivatives

$$\frac{\partial \mathbf{f}}{\partial \vartheta} = r \begin{pmatrix} \cos \vartheta \cos \varphi \\ \cos \vartheta \sin \varphi \\ -\sin \vartheta \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial \varphi} = r \begin{pmatrix} -\sin \vartheta \sin \varphi \\ \sin \vartheta \cos \varphi \\ 0 \end{pmatrix}$$

The metric coefficients read

$$g_{11} = r^2 \sin^2 \vartheta, \quad g_{12} = 0, \quad g_{22} = r^2, \quad (\text{D.3})$$

and the only non-vanishing Christoffel symbols of the second kind are

$$\Gamma_{12}^1 = \cot \vartheta, \quad \Gamma_{11}^2 = -\sin \vartheta \cos \vartheta. \quad (\text{D.4})$$

#### Appendix D.3. Ellipsoid

While the parametric representation of an ellipsoid is similar to the representation of a sphere, the metric coefficients are quite cumbersome. Again, we use spherical coordinates ( $u^1 = \varphi, u^2 = \vartheta$ ) but now  $\vartheta$  denotes the latitude angle,

$$\mathbf{f}(\varphi, \vartheta) = \begin{pmatrix} a \cos \vartheta \cos \varphi \\ b \cos \vartheta \sin \varphi \\ c \sin \vartheta \end{pmatrix}. \quad (\text{D.5})$$

The partial derivatives are straightforward:

$$\frac{\partial \mathbf{f}}{\partial \vartheta} = \begin{pmatrix} -a \sin \vartheta \cos \varphi \\ -b \sin \vartheta \sin \varphi \\ c \cos \vartheta \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial \varphi} = \begin{pmatrix} -a \cos \vartheta \sin \varphi \\ b \cos \vartheta \cos \varphi \\ 0 \end{pmatrix}.$$

The metric coefficients are straightforward and the Christoffel symbols of the second kind read

$$\begin{aligned}\Gamma_{11}^1 &= -\frac{(b^2 - a^2)c^2 \cos(\varphi) \sin(\varphi) \cos(\vartheta)^2}{\rho}, & \Gamma_{11}^2 &= \frac{a^2 b^2 \cos(\vartheta) \sin(\vartheta)}{\rho}, \\ \Gamma_{22}^1 &= -\frac{(b^2 - a^2)c^2 \cos(\varphi) \sin(\varphi)}{\rho}, & \Gamma_{12}^1 &= -\cot \vartheta, \\ \Gamma_{22}^2 &= \frac{[(b^2 - a^2)c^2 \sin(\varphi)^2 - b^2 c^2 + a^2 b^2] \cos(\vartheta) \sin(\vartheta)}{\rho}\end{aligned}$$

where  $\rho = a^2 b^2 \sin(\vartheta)^2 + [(a^2 - b^2)c^2 \sin(\varphi)^2 + b^2 c^2] \cos(\vartheta)^2$ .

#### Appendix D.4. Frustum

A frustum is defined by the upper and lower radii  $(r_1, r_2)$  and the height  $h$ . With parameters  $u^1 = \varphi \in [0, 2\pi)$  and  $u^2 = z \in [0, h]$ , we have

$$\mathbf{f}(\varphi, z) = \begin{pmatrix} \rho(z) \cos \varphi \\ \rho(z) \sin \varphi \\ z \end{pmatrix} \quad \text{with} \quad \rho(z) = r_1 - \frac{r_1 - r_2}{h} z. \quad (\text{D.6})$$

The partial derivatives are straightforward:

$$\frac{\partial \mathbf{f}}{\partial \varphi} = \begin{pmatrix} -\rho(z) \sin \varphi \\ \rho(z) \cos \varphi \\ 0 \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial z} = \begin{pmatrix} \rho'(z) \cos \varphi \\ \rho'(z) \sin \varphi \\ 1 \end{pmatrix},$$

where  $\rho'(z) = d\rho/dz = -(r_1 - r_2)/h$ . From these partial derivatives we obtain the surface normal

$$\mathbf{n}(\varphi, z) = \frac{1}{\sqrt{1 + \rho'(z)^2}} \begin{pmatrix} \cos \varphi \\ \sin \varphi \\ -\rho'(z) \end{pmatrix}. \quad (\text{D.7})$$

The metric coefficients read

$$g_{11} = \rho(z)^2, \quad g_{12} = 0, \quad g_{22} = \rho'(z)^2 + 1,$$

and the non-vanishing Christoffel symbols of the second kind are

$$\Gamma_{11}^2 = -\frac{(r_1 - r_2)^2 + h r_1 (r_2 - r_1)}{(r_1 - r_2)^2 + h^2}, \quad \Gamma_{12}^1 = \frac{r_2 - r_1}{(r_2 - r_1)z + h r_1}.$$

#### Appendix D.5. Torus

A torus is defined by two radii, where  $R$  is the radius of the main circle and  $r$  is the 'thickness' radius,

$$\mathbf{f}(\vartheta, \varphi) = \begin{pmatrix} (R + r \cos \vartheta) \cos \varphi \\ (R + r \cos \vartheta) \sin \varphi \\ r \sin \vartheta \end{pmatrix}. \quad (\text{D.8})$$

Here,  $u^1 = \vartheta \in [0, 2\pi)$  and  $u^2 = \varphi \in [0, 2\pi)$ , and the partial derivatives read

$$\frac{\partial \mathbf{f}}{\partial \vartheta} = r \begin{pmatrix} -\sin \vartheta \cos \varphi \\ -\sin \vartheta \sin \varphi \\ \cos \vartheta \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial \varphi} = \begin{pmatrix} -(R + r \cos \vartheta) \sin \varphi \\ (R + r \cos \vartheta) \cos \varphi \\ 0 \end{pmatrix}$$

The metric coefficients are straightforward

$$g_{11} = r^2, \quad g_{12} = 0, \quad g_{22} = (R + r \cos \vartheta)^2, \quad (\text{D.9})$$

and the non-vanishing Christoffel symbols of the second kind read

$$\Gamma_{12}^2 = -\frac{r \sin \vartheta}{R + r \cos \vartheta}, \quad \Gamma_{22}^1 = \frac{(R + r \cos \vartheta) \sin \vartheta}{r}. \quad (\text{D.10})$$

## References

- [1] J. J. Thomson. On the Structure of the Atom: an Investigation of the Stability and Periods of Oscillation of a number of Corpuscles arranged at equal intervals around the Circumference of a Circle; with Application of the Results to the Theory of Atomic Structure. *Philos. Mag.*, 7:237–265, 1904.
- [2] N. Ashby and W. E. Brittin. Thomson’s problem. *American Journal of Physics*, 54(9):776–777, 1986.
- [3] William T. Scott. Who Was Earnshaw? *Am. J. Phys.*, 27(6):418–419, 1959.
- [4] S. Earnshaw. On the Nature of the Molecular Forces which Regulate the Constitution of the Luminiferous Ether. *Trans. Cambridge Phil. Soc.*, 7:97–114, 1842.
- [5] L. L. Whyte. Unique arrangements of points on a sphere. *Am. Math. Month.*, 59(9):606–611, 1952.
- [6] E. Marx. Five charges on a sphere. *Journal of the Franklin Institute*, 290(1):71–74, 1970.
- [7] T. Erber and G. M. Hockney. Equilibrium configurations of N equal charges on a sphere. *Journal of Physics A: Mathematical and General*, 24(23):L1369, 1991.
- [8] Eric Lewin Altschuler and Antonio Pérez-Garrido. Global minimum for Thomson’s problem of charges on a sphere. *Phys. Rev. E*, 71:047703, 2005.
- [9] Eric Lewin Altschuler and Antonio Pérez-Garrido. Defect-free global minima in Thomson’s problem of charges on a sphere. *Phys. Rev. E*, 73:036108, Mar 2006.
- [10] Rainer Backofen, Axel Voigt, and Thomas Witkowski. Particles on curved surfaces: A dynamic approach by a phase-field-crystal model. *Phys. Rev. E*, 81:025701, Feb 2010.
- [11] H. Lakhbab, S. El Bernoussi, and A. El Harif. Energy minimization of point charges on a sphere with a hybrid approach. *Appl. Math. Sci.*, 6(30):1487–1495, 2012.
- [12] Thomson Problem @ S.U. <http://thomson.phy.syr.edu>.
- [13] OpenMP. <http://openmp.org>.
- [14] The OpenGL Utility Toolkit (GLUT) is a window system independent toolkit that delivers an application programming interface. <http://www.opengl.org/resources/libraries/glut>.
- [15] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.
- [16] Details to the Open Graphics Library (OpenGL) and the OpenGL Shading Language can be found at <http://www.opengl.org>.
- [17] L. Glasser and A. G. Every. Energies and spacings of point charges on a sphere. *Journal of Physics A: Mathematical and General*, 25(9):2473, 1992.
- [18] J. R. Morris, D. M. Deaven, and K. M. Ho. Genetic-algorithm energy minimization for point charges on a sphere. *Phys. Rev. B*, 53:R1740–R1743, Jan 1996.